IN THE CLAIMS

1 (Currently Amended). A method comprising:

writing a dirty cache line to a disk drive prior to a disk driver loading; ~~and~~

monitoring for a disk write request prior to said disk driver loading;

maintaining a write log for a process that directly accesses the disk drive;

utilizing said write log to update a disk cache and to maintain coherence with the disk drive; and

determining if the disk drive has a stored indication to indicate that a write log is maintained for said disk drive and said disk drive cache.

2 (Original). The method of claim 1 further comprising logging said disk write request if said disk driver has not loaded.

3 (Original). The method of claim 2 further comprising executing said disk write request.

4 (Original). The method of claim 1 further comprising refreshing said cache line after said disk driver is loaded.

5 (Original). The method of claim 1 further comprising monitoring for said write requests by executing code that is stored in a second memory.

6 (Original). The method of claim 1 further comprising monitoring for said write requests by executing code that provides an interface between a basic input output system and a read only memory.

7 (Original). The method of claim 6 further comprising acknowledging a filter function.

8 (Original). The method of claim 7 further comprising executing said second memory code to record information of a write to said disk drive.

9 (Original). The method of claim 8 further comprising executing said basic input output system code to execute said write requests.

10 (Original). The method of claim 1 further comprising monitoring for said write requests by executing code that modifies a stack.

11 (Original). The method of claim 10 further comprising determining a stack offset.

12 (Original). The method of claim 11 further comprising using said stack offset to return control to an option read only memory.

13 (Currently Amended). An article comprising a medium storing instructions, that if executed, enable a processor-based system to:

write a dirty cache line to a disk prior to a disk driver loading; ~~and~~

monitor for a write request, prior to said disk driver loading;

maintain a write log for a process that directly accesses the disk drive;

utilize said write log to update a disk cache and to maintain coherence with the disk drive; and

determine if the disk drive has a stored indication to indicate that a write log is maintained for said disk drive and said disk drive cache.

14 (Original). The article of claim 13 further storing instructions, that if executed, enable a processor-based system to log said write request if said disk driver has not loaded.

15 (Original). The article of claim 14 further storing instructions, that if executed, enable a processor-based system to execute said write request to said disk.

16 (Original). The article of claim 13 further storing instructions, that if executed, enable a processor-based system to refresh said cache line if said disk driver is loaded.

17 (Original). The article of claim 13 further storing instructions, that if executed, enable a processor-based system to monitor for said write request by executing code that is stored in an option read only memory.

18 (Original). The article of claim 13 further storing instructions, that if executed, enable a processor-based system to monitor for said write request by executing code that provides and interface for a basic input output system and an option read only memory.

19 (Original). The article of claim 13 further storing instructions, that if executed, enable a processor-based system to monitor for said write request by executing code that modifies a stack.

20 (Currently Amended). A system comprising:

a processor;

a disk drive coupled to said processor;

a disk cache coupled to said processor and said disk drive; and

at least one memory device coupled to said processor storing instructions that, if executed, enable said system to write a dirty cache line to said disk drive prior to loading a disk driver, ~~and to~~ monitor for a disk write request prior to loading said disk driver, maintain a write log for a process that directly accesses the disk drive, utilize said write log to update a disk cache and to maintain coherence with the disk drive, and determine if the disk drive has a stored indication to indicate that a write log is maintained for said disk drive and said disk drive cache.

21 (Original). The system of claim 20 wherein said at least one memory device stores instructions, that if executed, enable said system to log said disk write request if said disk driver has not loaded.

22 (Original). The system of claim 21 wherein said at least one memory device stores instructions, that if executed, enable said system to execute said disk write request.

23 (Original). The system of claim 20 wherein said at least one memory device stores instructions, that if executed, enable said system to a refresh cache line after said disk driver is loaded.

24 (Original). The system of claim 20 wherein said at least one memory device stores instructions, that if executed, enable said system to monitor for said write requests by executing code that is stored in an option read only memory.

25 (Original). The system of claim 20 wherein said disk cache comprises a polymer memory.

26 (Original). The system of claim 20 wherein said disk cache comprises ferroelectric polymer memory.

27 (Original). A method comprising acknowledging a filter function for a second memory.

28 (Original). The method of claim 27 further comprising sending disk drive identification data to code executing from said second memory.

29 (Original). The method of claim 27 further comprising executing code from said second memory to write to a disk drive.

30 (Original). The method of claim 27 further comprising initializing a second memory as a drive request handler.

31 (Original). The method of claim 27 wherein said second memory further comprises an option read only memory.

32 (Original). The method of claim 30 further comprising determining a stack offset.

33 (Original). The method of claim 30 further comprising using said stack offset to return control to said second memory.